**Question 1:**

The first question asked us to generate {wi, xi} weighted pairs using 6 different N-values (17, 33, 65, 129, 257, 513) for all 3 Newton-Cotes methods:

1. Composite Midpoint rule (QuadSchemeMidpoint.m)
2. Composite Trapezoid rule (QuadSchemeTrapezoid.m)
3. Composite Simpson formula (QuadSchemeSimpson.m)

Implementation of the functions to get wi, xi values were more-or-less the same for all three methods. Vectors of size-N were pre-generated to hold the wi, xi, then Δx was calculated. Finally, a loop was used to handle generating {wi, xi } pairs with respect to N-number of partitions. Edge cases were handled outside the loop. The equations that were used to calculate the values for said pairings were obtained from the Numerical Integration slides presented in class.

This question did not ask for results and figures to be generated.

**Question 2:**

The Gaussian-Legendre was created via mapping the {wi, xi} pairs that were generated for [-1 1] to [a b]. The pairings that were produced were hard-coded and the value of N determined which sets of pairing was used via a switch statement. Then, like the methods in Question 1, the formula for Gauss was borrowed from the slides and implemented.

As with question 1, this question did not ask for results and figures to be generated.

**Question 3:**

Simpson’s formula converged the fastest of the three Newton-Cotes methods. This is probably due to the Trapezoidal rule being an average of the left and right Riemann Sums; as a result, the Composite Trapezoid Rule performs poorly when it comes to approximating very curvy equations, like ours which involves a few sine and cosines.

Simpson’s Rule, however, places a parabola across every two subintervals; this ends up being more complex and yields a more accurate result in our case.